

Analytický nástroj pro TeamSpeak

Analytic Tool for TeamSpeak

Zadání diplomové práce

Student:

Bc. Tomáš Juříčka

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Analytický nástroj pro TeamSpeak
Analytic Tool for TeamSpeak

Zásady pro vypracování:

Cílem práce je vytvořit analytický nástroj pro službu TeamSpeak, sloužící pro kolektivní komunikaci. Výsledkem by mělo být usnadnění správy TeamSpeak serverů pro jejich správce. Výsledná aplikace by měla splňovat následující požadavky: Aplikace typu klient-server, lokalizace do více jazyků (minimálně CZ/EN), Zabezpečená komunikace v Client-Server prostředí, hledání klientů s nestabilním připojením, hledání podobných uživatelů na základě podobnosti jmen a IP adres, analýzu chování jednotlivých klientů, detekce nevhodného chování klientů, zálohování a obnova nastavení s možností vytváření nových instancí dle požadavků.

Práce bude obsahovat:

1. Analýzu systému Team Speak.
2. Návrh implementace.
3. Implementace nástroje.
4. Otestování funkčnosti.

Seznam doporučené odborné literatury:

- [1] Jay Glynn a kol.: C# Programujeme profesionálně, COMPUTER PRESS, ISBN: 9788025100851
[2] Intelligent Data Analysis: An Introduction, Michael R. Berthold, David J. Hand, Springer, 2007, ISBN 978-3-540-43060-5

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Jan Platoš, Ph.D.**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 25. dubna 2013

.....*Juníčka*.....

Rád bych na tomto místě poděkoval všem administrátorům a uživatelům serveru `teamspeak4fun.eu`, kteří mi s prací pomohli a podpořili mě v testování programu.

Abstrakt

Tato práce si klade za cíl navrhnout, napsat a otestovat program, který usnadní práci administrátorům TeamSpeak3 serverů.

Klíčová slova: TeamSpeak3, automat, administrátor, nástroj

Abstract

The goal of this thesis is to propose, write and test a programme, that will be helpful for TeamSpeak3 server administrators.

Keywords: TeamSpeak3, bot, admin, tool

Seznam použitých zkratk a symbolů

cgid	– ID kanálové skupiny
cid	– ID kanálu
clid	– klientské ID
cluid	– klientské unikátní ID
cldbld	– klientské databáze ID
db	– databáze
gui	– grafické uživatelské rozhraní
os	– operační systém
pid	– ID pravomoci
sid	– ID virtuálního serveru
sgid	– ID serverové skupiny
ts3	– TeamSpeak3
vm	– virtuální stroj

Obsah

1	Úvod	6
2	TeamSpeak3	7
2.1	Server	7
2.2	Client	7
2.2.1	GUI	7
2.2.1.1	Menu	7
2.2.1.2	Rychlé volby	9
2.2.1.3	Serverové záložky	9
2.2.1.4	Uživatelé a místnosti	9
2.2.1.5	Detail	9
2.2.1.6	Textový výstup	9
2.2.1.7	Záložky chatů	9
2.2.1.8	Uživatel	9
2.2.1.9	Místnost	9
2.2.1.10	ServerQuery uživatel	9
2.3	ServerQuery	10
2.3.1	Používané zkratky/parametry	10
2.3.1.1	cid	10
2.3.1.2	clid	10
2.3.1.3	cluid	10
2.3.1.4	cldbld	10
2.3.1.5	msg	11
2.3.2	Nejpoužívanější příkazy	11
2.3.2.1	Quit	11
2.3.2.2	Login	11
2.3.2.3	Use	11
2.3.2.4	Servernotifyregister	11
2.3.2.5	Sendtextmessage	11
2.3.2.6	Logview	12
2.3.2.7	Messageadd	12
3	BotAdmin	13
3.1	Návrh implementace	13
3.2	O programu	17
3.3	Implementace	17
3.3.1	TsQueryAnswer	17
3.3.2	LogCheck	17
3.3.3	RemoteAccess	20
3.3.4	prepare.sql	20
3.3.4.1	Actions	20
3.3.4.2	channelConnected	21

3.3.4.3	IpAddress	21
3.3.4.4	Nicks	21
3.3.4.5	serverInfo	21
3.3.4.6	rules	22
3.3.4.7	timedRules	22
3.3.4.8	channelConnectedRule	22
3.3.4.9	pmCanIssue	22
3.3.5	SQLReader	22
3.3.6	Utils	23
3.3.7	TsQuery	23
3.3.8	TsQueryReader	23
3.4	Podmínky běhu	25
3.5	Instalace	25
3.6	Ovládání	25
3.6.1	Editace pravidel	30
3.6.1.1	Speciální příkazy	30
3.6.1.2	Speciální parametry příkazů	32
3.6.1.3	Časovaná pravidla	33
3.6.1.4	Pravidla založená na akcích	34
3.6.1.5	Pravidla založená na připojení uživatele do kanálu.	34
3.6.2	Příklady pravidel	34
3.6.2.1	Časovaná pravidla	34
3.6.2.2	Pravidla založena na akcích	35
3.6.2.3	Pravidla založena na připojení uživatele do kanálu	35
3.7	BotAdmin.remoteClient	36
3.8	Testovací provoz programu	37
4	Závěr	38
5	Reference	40

Seznam tabulek

1	Tabulka escape znaků	31
2	Příklady časovaných pravidel	34
3	Příklady pravidel založených na akcích	35
4	Příklady pravidel založených na připojení uživatele do kanálu	35

Seznam obrázků

1	GUI TS3 Client	8
2	Návrh implementace - use case	13
3	Sekvenční diagram - získání informací o klientovi	14
4	Sekvenční diagram - získání informací o klientovi 2	14
5	Sekvenční diagram - načítání dat do DB	15
6	Diagram aktivit - vyhledání duplicitních registrací	16
7	BotAdmin - hlavní okno	25
8	BotAdmin - nabídka Soubor	26
9	BotAdmin - úprava záložek	26
10	BotAdmin - záložky	27
11	BotAdmin - spuštěné připojení na TS3 server	27
12	BotAdmin - úprava dat v databázi	28
13	BotAdmin - detail uživatele	29
14	BotAdmin.remoteClient - editace záložek	36

Seznam výpisů zdrojového kódu

1	LogCheck - utilize_lines	19
2	TsQueryReader - načítání dat	24

1 Úvod

Cílem práce je navrhnout, napsat a otestovat program, který by správcům TS3 serverů usnadnil správu a management programu. Program by měl usnadnit správu uživatelů na TS3 serveru, vyhledávání informací o uživatelích a ulehčit či automatizovat ovládání a konfiguraci samotného TS3 serveru.

V sekci 2 TeamSpeak3 je popsán samotný program TeamSpeak3, k čemu slouží a jeho základní vlastnosti. Následuje sekce 3 BotAdmin, ve které je popsán návrh a popis struktury tvořeného programu, jeho vlastnosti, podmínky běhu a popis jeho ovládání.

Program byl pojmenován BotAdmin. Program vznikl z potřeby dohledávání informací o klientech, kteří způsobují problémy na TS3 serveru. V TS3 lze udělit bany, stížnosti, zablokovat přístupy a další. Jakmile ban vyprší, nedojde nikde k jeho archivaci (kromě logu). Lze jej tak dohledat pouze v logu. Obdobně to je u stížností a dalších akcí. Pokud uživatel/administrátor provede některou z akcí (přidání uživatele do skupiny, editace serveru atd.), tak se tyto informace (kdy, kde, kdo) složitě dohledávají. Přesto, že se tyto informace ukládají do log souboru, TS3 client má omezené možnosti jejich dohledání. Lze filtrovat různé levely logu, ale vždy se načítá po 100 řádcích, tzn. při lehkém provozu na serveru cca 4 hodiny dat. Mezi záznamy lze vyhledávat pouze fulltextově a vyhledání informací o jediném klientovi se tak stává časově náročné (hodiny až dny). TS3 také neumožňuje automatizovat některé akce na serveru, které by administrátorům usnadnily správu serveru.

Vytvořený program vyřeší výše zmíněné nedostatky, přidá funkce pro hromadné posílání zpráv do skupin a ukládání/obnovu konfigurace serveru ze souboru. Program vytvoří interface, díky kterému lze automatizovat vykonávané operace na serveru. Program zároveň vytvoří univerzální rozhraní, které dovolí využít všech příkazů TS3 ServerQuery a přidá příkazy, které nejsou na TS3 implementovány.

2 TeamSpeak3

TeamSpeak3 je program, který umožňuje textovou a hlasovou komunikaci přes internet, případně v lokální síti. Program zároveň umožňuje přenos souborů a skládá se ze dvou částí:

1. Server
2. Client

Jednou z výhod programu je, že funguje napříč různými platformami. Serverovou i klientskou část lze používat na systémech Windows, Linux i Mac OS X. Navíc klienta lze používat na iOS a Android.

2.1 Server

TS3 Server je serverová část programu, která se spouští na serveru. Neobsahuje GUI a ovládá se přes TS3 ServerQuery nebo TS3 Klienta. Při prvním spuštění se vytvoří jeden virtuální server s 32 sloty a vygeneruje se jeden token pro získání pravomocí na serveru po připojení přes TS3 Klienta. Zároveň se vytvoří "super uživatel". Tento "super uživatel" bude pojmenován serveradmin a je poskytnuto heslo. Slouží k přístupu přes TS3 ServerQuery. Má plná práva a může kompletně spravovat server.

2.2 Client

TS3 Client je klientská aplikace, umožňující připojení klienta na libovolný počet serverů.

2.2.1 GUI

Na různých platformách vypadá GUI různě, ale funkcionalitu mají stejnou. Taktéž vzhled se může lišit od použité verze a skinu programu. Zde si popíšeme klienta ve verzi pro OS Windows. Klient je v anglické jazykové mutaci, má základní skin a je ve verzi 3.0.9.2.

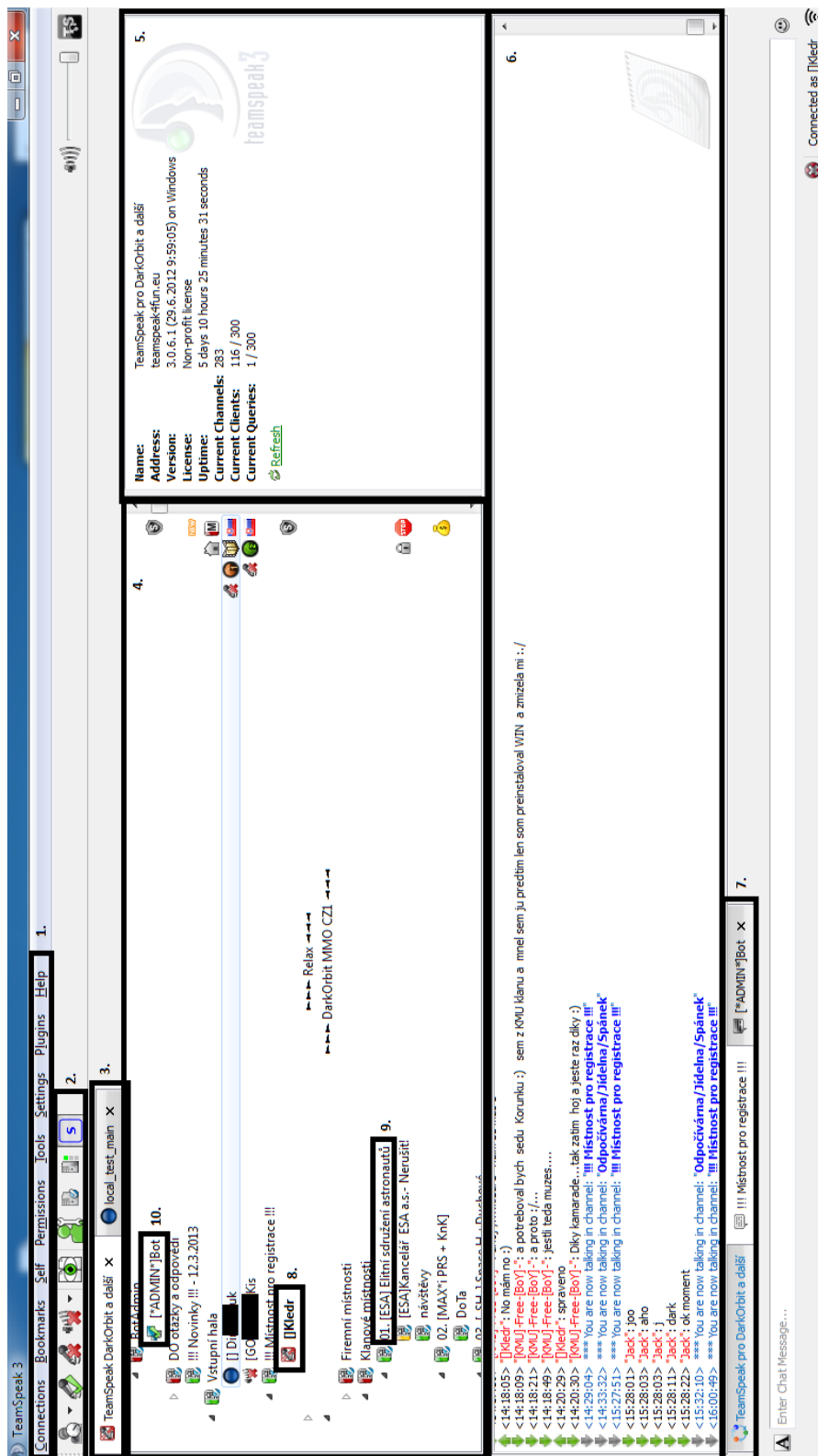
Na obrázku 1 je zobrazeno toto GUI a jsou zde také označeny jednotlivé sekce, které si dále popíšeme.

2.2.1.1 Menu V rozevíracím seznamu *Connections* máme možnost k připojení na server, odpojení z aktuálně zvoleného serveru, všech serverů a ukončení celého programu.

V rozevíracím seznamu *Bookmarks* máme možnost spravování záložek a připojení přes předem uložené záložky na různé servery.

V rozevíracím seznamu *Self* máme možnost zvolení profilů pro komunikaci přes TS3 (nastavení mikrofону, repro atd.), změnu přezdívky, informací o spojení a úpravu avataru.

Rozevírací seznam *Permissions* je určen pro administrátory serveru. Lze zde spravovat serverové skupiny, kanálové skupiny, jejich pravomoci a pravomoci jednotlivých uživatelů.



Obrázek 1: GUI TS3 Client

V rozevíracím seznamu *Tools* najdeme různé užitečné informace, jak pro obvyčejné uživatele, tak pro administrátory. Najdeme zde seznamy stížností, banů, přátel, offline zpráv, logy

V rozevíracím seznamu *Settings* nalezneme zprávu identit, pluginů klienta a detailní nastavení klienta.

2.2.1.2 Rychlé volby Zde si lze umístit rychlé volby pro nastavení Away statusu, aktivaci/deaktivaci mikrofonu, zvukového výstupu

2.2.1.3 Serverové záložky Pokud jsme připojeni na více než jeden server, tak se zde zobrazí záložky jednotlivých serverů.

2.2.1.4 Uživatelé a místnosti Stromovou strukturou jsou zde řazeny místnosti a uživatelé, kteří jsou připojeni do této místnosti. V místnosti může být více uživatelů a také zde mohou být podmístnosti.

2.2.1.5 Detail Zde se zobrazují detailnější informace o uživateli nebo místnosti zvolené v 2.2.1.4.

2.2.1.6 Textový výstup Okno, kde se zobrazuje textový výstup zvolené záložky 2.2.1.7.

2.2.1.7 Záložky chatů Jako první je vždy záložka serveru. Zobrazují se tu informace o pohybu klientů, o modifikacích místností, serveru a další informace týkající se serveru. Jako druhá je vždy záložka odpovídající chatu v dané místnosti. Zobrazují se zde textové zprávy jednotlivých uživatelů, kteří jsou v místnosti a píšou do chatu této místnosti. Následně se řadí soukromé chaty mezi uživateli celého serveru.

2.2.1.8 Uživatel Uživatel připojený do určité místnosti. Lze ho přes menu kontaktovat (jak textově tak hlasově), přetažením myši přesouvat a provádět další operace.

2.2.1.9 Místnost Místnost do které se mohou uživatelé připojovat, lze upravovat popis a další parametry místnosti.

2.2.1.10 ServerQuery uživatel Je identifikován speciální ikonkou před přezdívkou. Nelze s ním hlasově ani textově komunikovat. Textově lze komunikovat pouze v případě, že se zaregistroval pro textový chat.

2.3 ServerQuery

Je to interface (telnet) sloužící k ovládání a komunikaci s TS3 Serverem pomocí textových příkazů. Kompletní seznam příkazů a omezení kladená na tento interface lze nalézt v dokumentu [1].

Komunikace po tomto spojení včetně přenosu hesla probíhá nezabezpečeně. Případnému útočníkovi na server stačí odchytnout heslo a získá plnou kontrolu nad TS3 Serverem.

Po připojení na interface se server identifikuje jako TS3 Query a připojený klient je v serverové skupině Guest Server Query. Lze okamžitě posílat příkazy. Po autentizaci příkazem login a use již máme pravomoci stejné, jako by jsme byli připojeni přes TS3 Klienta.

Při komunikaci s interface se jako dělicí znak mezi parametry příkazů používá mezera. Proto při posílání textu a některých parametrů musí být mezera a pár dalších speciálních znaků nahrazeno. Toto je popsáno v dokumentu [1] v sekci Escaping a zkráceně v sekci 3.6.1.1.6.

2.3.1 Používané zkratky/parametry

TS3 má zavedeno pár zkratk, resp. parametrů pro TS3 ServerQuery. Na základě osobní zkušenosti používání a programování tohoto programu jsem zjistil, že označení nejsou vždy jednotná a lze se setkat s různými variantami značení.

2.3.1.1 cid Jedná se číselné označení kanálu. Každý kanál při vytvoření dostane unikátní ID. Toto ID se v celém programu nikdy neopakuje, tzn. při vytvoření nového virtuálního serveru se ID negeneruje od jedničky. Také při obnovení TS3 ze zálohy jsou ID inkrementovaná, tzn. jakoby kanály byly všechny odstraněny a vytvořeny nové.

2.3.1.2 clid ID, které dostane každý uživatel při připojení na server. V jednom okamžiku nemůžou být dva uživatelé se stejným clid na serveru. Jakmile se ale uživatel odpojí, tak je clid uvolněno a po nějaké době opět vráceno do poolu. Poté může být přiděleno jinému uživateli.

Můžeme se také setkat s označením *invokerid*, *targetid*, *target*, *invoker*.

2.3.1.3 cluid ID, které vygeneruje TS3 Klient při svém prvním spuštění. Jedná se o hash textového řetězce, který je vygenerován klientem. Jednoznačně identifikuje uživatele TS3 napříč TS3 servery po celém světě. Toto cluid může být vyexportováno a přeneseno na jiný systém, PC atd., kde jej lze importovat.

Můžeme se také setkat s označením *client_unique_identifier*, *invokeruid*, *targetuid*.

2.3.1.4 cldbid ID(číslo) přiřazeno každému uživateli, který se poprvé připojí s novým cluid na virtuální server. Tato ID jsou automaticky generována a nemůže se stát, že by na dvou virtuálních serverech na jednom TS3 serveru byla dvě stejná cldbid.

2.3.1.5 msg Text zprávy, kterou lze zaslat pomocí některého z příkazů.
Můžeme se také setkat s označením *message*.

2.3.2 Nejpoužívanější příkazy

Zde jsou nejpoužívanější příkazy, které budu používat ve mnou vytvořeném programu a jejich popis.

2.3.2.1 Quit Ukončí spojení na TS3 ServerQuery.

2.3.2.2 Login Dojde k autentizaci s instancí TS3 serveru. Přihlašovací údaje si každý uživatel, pokud má na to pravomoci, může vygenerovat v TS3 Clientovi.

Syntaxe:

```
login client_login_name={username} client_login_password={password}
login {username} {password}
```

2.3.2.3 Use Dojde ke zvolení virtuálního serveru a klient se objeví ve výchozí místnosti všem uživatelům, kteří mají pravomoci a zapnuté zobrazení query klientů. Lze se přihlašovat pomocí serverId nebo pomocí portu, na kterém je server provozován.

Syntaxe:

```
use sid={serverID}
use port={serverPort}
```

2.3.2.4 Servernotifyregister Pokud se klient nezaregistruje pro žádné eventy, tak ze strany serveru nebude iniciována komunikace. Klient pouze dostane odpověď na dotaz /-příkaz, který odeslal serveru. Když se klient například zaregistruje pro "textprivate" bude klient schopen přijímat privátní zprávy. Pokud někdo pošle zprávu našemu klientovi dostane od serveru řádek "notifytextmessage. . ." s parametry a odpovídající zprávou.

Syntaxe:

```
servernotifyregister      event={server|channel|textserver|textchannel|textprivate}
[id={channelID}]
```

2.3.2.5 Sendtextmessage Odešle textovou zprávu dle parametru targetmode:

1. Cílem je klient
2. Cílem je kanál(musí být v tomto kanálu připojen)
3. Cílem je virtuální server (musí být na tomto serveru připojen - use)

ClientID v tomto příkazu je clid, nikoli cldbid nebo cluid.

Syntaxe:

```
sendtextmessage      targetmode={1-3}      target={serverID|channelID|clientID}
msg={text}]
```

2.3.2.6 Logview Příkaz, díky němuž můžeme číst záznamy z posledního vytvořeného log souboru. Parametr *lines* udává kolik řádků bude načteno z logu. Maximální počet načtených řádků jedním dotazem je 100. Reverse určuje, zda budou seřazeny vzestupně nebo sestupně podle data. Pokud bude instance nastavena, nebude se číst z logu virtuálního serveru, ale z master logu (tj. logu celého serveru). Begin_pos značí odkud se bude číst. Načte ze vždy od pozice <begin_pos> <lines> starších řádků.

Odpověď vypadá: *last_pos=403788 file_size=411980 l=p\slisting. . .*

last_pos značí polohu posledního znaku, který byl z log souboru načten.

file_size značí velikost log souboru.

Tyto informace lze použít k načtení následujících řádků z logu.

Syntaxe:

logview [lines={1-100}] [reverse={1|0}] [instance={1|0}] [begin_pos={n}]

2.3.2.7 Messageadd Odešle offline zprávu klientovi specifikovaného pomocí cluid. Předmět zprávy specifikovaný v subject a text zprávy v message.

Syntaxe:

messageadd cluid=clientUID subject=subject message=text

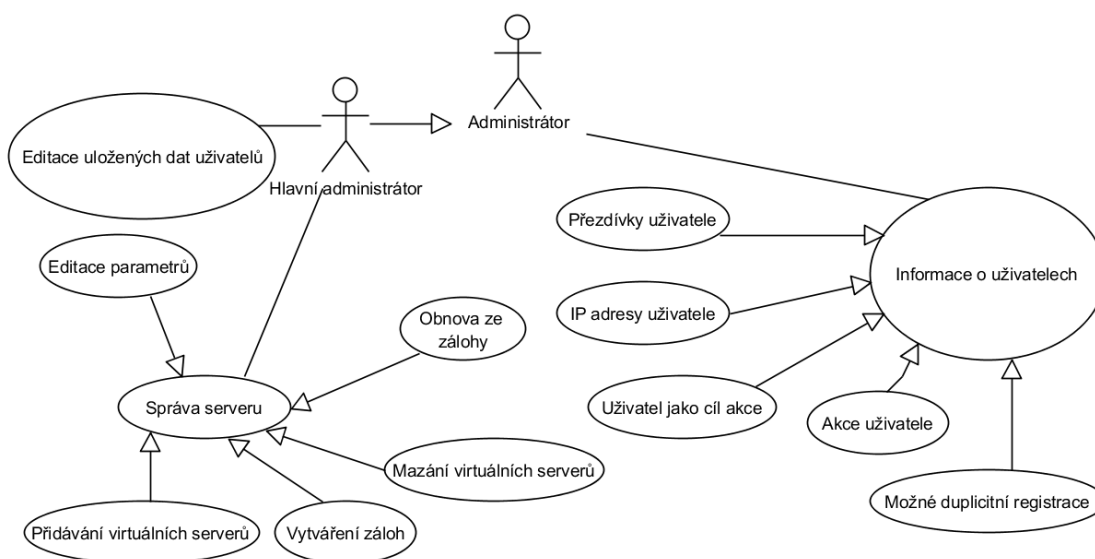
3 BotAdmin

Vytvořený program jsem pojmenoval BotAdmin. Program je naprogramován v jazyce JAVA. K programování byl použit editor NetBeans IDE ve verzi 7.2. GUI programu je vytvořeno integrovaným layout managerem s využitím knihovny Swing. Program byl vytvořen pro TS3 server ve verzi 3.0.6.1. Měl by být kompatibilní i se staršími verzemi a pokud nedojde ke změnám v TS3 ServerQuery bude kompatibilní i s novějšími.

Program je navržen k běhu na serveru společně s TS3 serverem. Je schopen běžet i jinde na síti, ale vzhledem k objemu dat, neustálé komunikaci a nemožnosti zabezpečení spojení s TS3 Serverem to není vhodné. Dále je vytvořen program BotAdmin_remote, který umožňuje vzdálené zabezpečené připojení na spuštěný program BotAdmin. Umožňuje vyhledávání informací o klientech, ale neumožňuje upravovat databázi. Proto nelze přes vzdálený přístup přidat automatická pravidla.

3.1 Návrh implementace

Na obrázku 2 je znázorněn diagram případů užití pro BotAdmin. Jsou předpokládány dvě role. Běžný administrátor a pak hlavní administrátor, který má pravomoci běžného administrátora a také možnost správy samotného serveru.



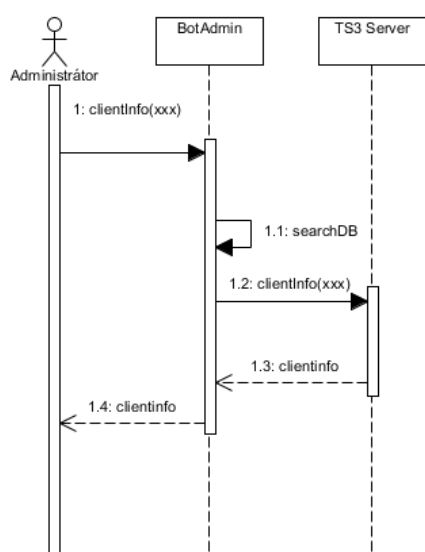
Obrázek 2: Návrh implementace - use case

Běžný administrátor má možnost nahlížet do dat jednotlivých uživatelů. Předpokládá se zaznamenávání akcí uživatelů, použité přezdívky uživatelů a jejich četnost a zaznamenávání IP adres uživatelů a četnost připojení z těchto adres. Na základě těchto uložených informací lze poté vyhledávat duplicitní registrace uživatelů.

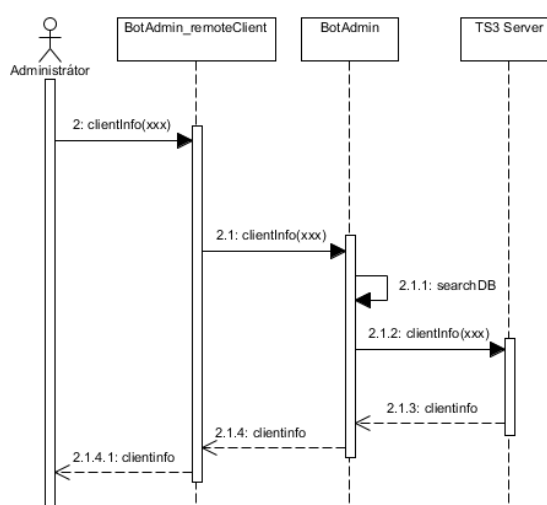
Hlavní administrátor by měl mít možnost přidání/editaci/smazání dat z databáze, kde jsou uložena data o uživatelích. Dále musí mít možnost správy TS3 Serveru. Předpokládá se možnost úpravy instancí TS3 serverů, vytváření nových instancí serveru, mazání instancí serverů, vytváření záloh serverů a jejich obnova z těchto záloh.

Z tohoto diagramu a dalších požadavků tedy vyplývají následující **funkcionální požadavky**:

- Vyhledání informací o klientech pro možnost analýzy chování.
- Vyhledávání duplicitních registrací klientů.
- Záloha instancí virtuálních serverů.
- Obnova instancí virtuálních serverů.
- Vytváření instancí virtuálních serverů.
- Mazání instancí virtuálních serverů.
- Zabezpečený vzdálený přístup k informacím o klientech.
- Různé jazykové mutace GUI programu.
- Ukládání získaných dat pro pozdější vyhledání/zpracování.



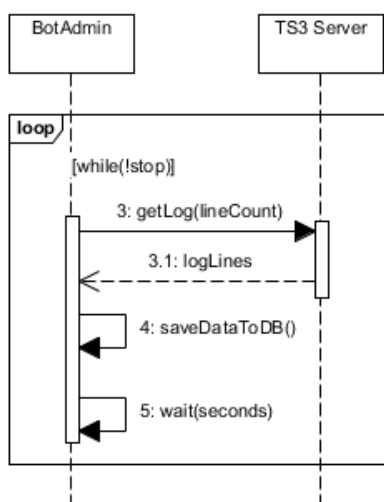
Obrázek 3: Sekvenční diagram - získání informací o klientovi



Obrázek 4: Sekvenční diagram - získání informací o klientovi 2

Na obrázku 3 a 4 je zobrazen sekvenční diagram pro dotaz na informace o klientovi. Administrátor zadá dotaz na klienta programu BotAdmin. BotAdmin prohledá svou databázi, doptá se na další informace přímo TS3 Serveru a tyto dohledané informace zobrazí

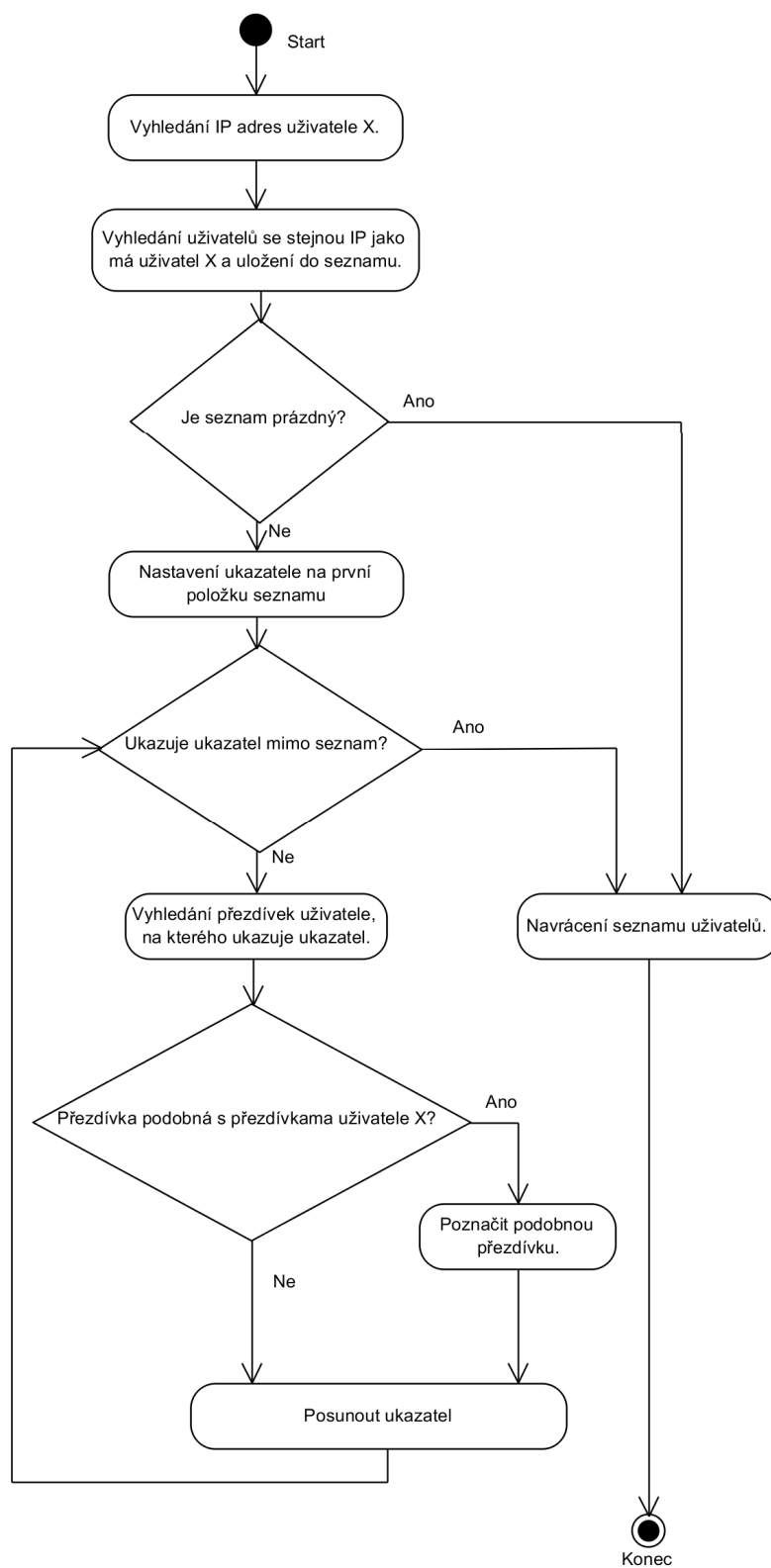
administrátorovi. V obrázku 4 je zobrazen mezikrok popisující postup administrátora, který nepokládá dotaz přímo programu BotAdmin. Administrátor zadá požadavek na vzdáleného klienta BotAdmin, ten předá požadavek na program BotAdmin a ten dále pokračuje stejně jako ve výše zmíněném příkladu. BotAdmin_remoteClient je pouze zprostředkovatelem pro předávání informací po zabezpečeném spojení. Sám nevyhledává v DB ani aktivně nevyvolává komunikaci.



Obrázek 5: Sekvenční diagram - načítání dat do DB

Na obrázku 5 je sekvenční diagram zobrazující načítání dat ze záznamů TS3 serveru. Protože na TS3 Query nelze nastavit, aby potřebné informace samo posílalo, je třeba periodicky načítat záznamy TS3 Serveru. Tyto záznamy se roztřídí, analyzují a uloží do vlastní DB pro pozdější použití.

Na obrázku 6 je zobrazena aktivita vyhledání duplicitních registrací. Nejdříve se vyhledají klienti, kteří se někdy připojili ze stejných IP adres jako sledovaný klient X (aspoň z jedné). Pokud se žádný klient nenajde, vrátí se prázdný seznam. Pokud se klienti najdou, tak se nastaví ukazatel na prvního klienta v seznamu. Následně se postupně porovnají přezdívky každého z dohledaných klientů s přezdívkami klienta X. Pokud se nalezne shoda (aspoň částečná), zaznačí se dohledaná přezdívka. Nakonec se navrátí seznam s uživateli a případnou poznámkou ve shodě přezdívkou.



Obrázek 6: Diagram aktivit - vyhledání duplicitních registrací

3.2 O programu

Program jsem se rozhodl psát metodou extrémního programování. Požadavky na program jsou dány zadáním, ale postupem času a ze zpětné vazby při testování se program stával univerzálnějším. Ze začátku to vypadalo, že se program obejde bez databáze a bude dostačující ukládat data do jednoduchých souborů. Nicméně při následném rozšiřování programu se ukázalo, že bude třeba použít databáze.

Rozhodl jsem se pro SQLite databázi. Databáze je jednoduchá (jeden soubor) a nepotřebuje instalaci. Zvolením této databáze došlo ke zjednodušení programového kódu a zároveň databáze umožnila rychlejší práci s daty ukládanými programem. Protože tato databáze dodržuje SQL standard lze v budoucnu v případě potřeby přejít s lehkými úpravami na MySql či jiný druh SQL databáze.

Program je od začátku projektován, aby běžel na stejném systému jako TS3 server. Aby byl umožněn přístup více uživatelům/administrátorům najednou a nemusel být dáván přístup k serveru, byl vytvořen vzdálený klient. Tento klient umožňuje nahlížet do databáze a vyhledávat potřebné informace o klientech. Také byl vytvořen přístup přes TS3 klienta. Ve své podstatě se jedná o chat s TS3 query uživatelem. Tento query uživatel je v základu program BotAdmin připojený na server. Většinu příkazů tak lze přímo provádět přes chat s tímto uživatelem.

Pro program jsem vygeneroval JAVA dokumentaci, kde je popsáno, jakou mají jednotlivé třídy a metody funkci. Také jsou zde umístěny UML diagramy tříd. Tato dokumentace je uložena v příloze na CD. Důležitější metody a třídy jsou popsány v následující sekci 3.3.

3.3 Implementace

Kompletní diagram tříd nebylo možno vzhledem k velikosti obrázku vložit do textu práce. Lze jej tedy najít v příloze na CD.

3.3.1 TsQueryAnswer

Třída, do které se ukládají data, které přijdou jako odpověď na příkaz z TS3 ServerQuery. Query posílá jako odpověď vždy dva řádky. V prvním řádku je samotná odpověď, která může mít od nula až po několik tisíc znaků. Většinou obsahuje daný řádek max 200 znaků. Na druhém řádku je error status odpovědi (error id=0 msg=ok). Pokud je vše v pořádku, je ID odpovědi 0 a textové vyjádření ok. Pokud dojde k nějaké chybě, je zde uvedeno ID chyby a krátké textové vyjádření popisující chybu.

3.3.2 LogCheck

Tato třída se stará o načtení záznamů TS3 serveru přes TS3 ServerQuery. Tyto záznamy postupně analyzuje a uloží do databáze programu BotAdmin. Protože jsem v dokumentaci k TS3 serveru nikde nenašel, jaké záznamy se v logu serveru ukládají, bylo třeba projít logy serveru a nalézt jednotlivé druhy záznamu. Pár záznamů týkajících se pouze uživatelů je uvedeno níže.

- 2011-11-11 04:02:53.869395 | INFO | VirtualServer | 1 | client connected '[S.A]pítrš_19_LVL'(id:3329) from xxx.xx.xx:xxxx
- 2011-11-11 04:16:06.845395|INFO |VirtualServer | 3| client disconnected '[«ČL»]Bi-gSchock'(id:2766) reason 'reasonmsg=leaving'
- 2011-11-11 04:07:19.594395|INFO |VirtualServer | 1| client disconnected '[EBTX]Predy'(id:1787) reason 'reasonmsg=connection lost'
- 2011-11-11 06:23:46.642395|INFO |VirtualServer | 1| channel '22.[T C]Tetčice Computer'(id:906) edited by '[exe]†ČističOrbitu•17lvI'(id:2112)
- 2011-11-12 18:49:38.846395|INFO |VirtualServer | 1| client '[EN]Kenny54'(id:3518) was added to channelgroup 'Channel Admin'(id:5) by client '[EN]Kotatko'(id:2445) in channel '21.[EN] Elitní Nervaci'(id:905)
- 2011-11-13 08:58:30.869395|INFO |VirtualServer | 1| client disconnected '[TFA]Sorat Kolka'(id:3289) reason 'invokerid=0 invokename=server reasonmsg=idle time exceeded'
- 2011-11-12 17:26:12.275395|INFO |VirtualServer | 1| client disconnected '[TFA]Sorat Kolka'(id:2446) reason 'invokerid=109 invokename=[]micro invokeruid=eFzVbV4zhNPxevDqHISsDYoln6I= reasonmsg=di pric'
- 2011-11-12 17:28:06.896395|INFO |VirtualServer | 1| client disconnected '[TFA]Sorat Kolka'(id:2446) reason 'invokerid=109 invokename=[]micro invokeruid=eFzVbV4zhNPxevDqHISsDYoln6I= reasonmsg=nevhodné chování bantime=3600'
- 2011-11-13 20:13:14.192395|INFO |VirtualServer | 1| permission 'i_channel_needed_modify_power'(id:13170) with values (value:20) was added by '[ESA]Ozmejmr'(id:180) to channel '[VK] Vesmírní Kovbojové'(id:1096)

Toto je pouze skromný výčet záznamů, které je třeba analyzovat, vyčíst z nich potřebné údaje a uložit si je. Ve třídě je implementován záznam řádků, při jejichž analýze došlo k chybě a taky záznam řádků, které dosud nebyly implementovány. Postupem času při testovacím provozu se třída rozrůstala a bylo doplňováno třídění informací.

Třída cyklicky čte řádky, které načte z TS3 ServerQuery. Tyto řádky zpracuje metodou *utilize_lines*. Metoda řádky rozparsuje a uloží data do databáze a navrátí poslední uložený řádek. Toto je důležité pro následný běh, aby nedošlo k duplicitnímu zpracovávání řádků. Následuje část kódu, na kterém je vysvětleno, jak se zpracovávají řádky z logu.

```

private String  utilize_lines (String lines) throws SQLException {
    ...
    tmp = lineArray[i].substring("l=".length());
    tmpSplit = tmp.split("\\\\p", 5);
    ...
    date = formatter.parse(Utils.unescape(tmpSplit[0])).getTime();
    ..
    serverId = Integer.parseInt(tmpSplit[3]);
    ...
    lineParser = tmpSplit[4].substring(2).split("\\\\s", 2);
    switch (lineParser[0]) {
        case "client":
            lineParser = lineParser[1].split("\\\\s", 2);
            switch (lineParser[0]) {
                case "connected":
                    temp1 = lineParser[1].indexOf('\'');
                    temp2 = lineParser[1].indexOf("'id:", temp1);
                    nick = Utils.toReadable(lineParser[1].substring(temp1 + 1, temp2));
                    temp1 = lineParser[1].indexOf(")", temp2);
                    try {
                        cldbid = Integer.parseInt(lineParser[1].substring(temp2 + 5, temp1));
                    } catch (NullPointerException | NumberFormatException numberFormatException)
                    {
                        cldbid = 0;
                    }
                    temp2 = lineParser[1].indexOf("\\sfrom\\s", temp1);
                    temp1 = lineParser[1].indexOf(':', temp2);
                    ip = lineParser[1].substring(temp2 + 8, temp1);
                    virtualServer.getDbQuery().insertNick(nick, cldbid);
                    virtualServer.getDbQuery().insertIp(ip, cldbid);
                    break;
                case "disconnected":
                    temp1 = lineParser[1].indexOf('\'');
                    temp2 = lineParser[1].indexOf("'id:", temp1);
                    nick = Utils.toReadable(lineParser[1].substring(temp1 + 1, temp2));
                    temp1 = lineParser[1].indexOf(")", temp2);
                    try {
                        cldbid = Integer.parseInt(lineParser[1].substring(temp2 + 5, temp1));
                    } catch (NullPointerException | NumberFormatException numberFormatException)
                    {
                        cldbid = 0;
                    }
                    virtualServer.getDbQuery().insertNick(nick, cldbid);

                    break;
                default:
                    ...
            }
        }
    }
}

```

Výpis 1: LogCheck - utilize_lines

1. Nejdříve je odstraněn začátek řádku při načtení přes TS3 ServerQuery.
2. Poté je záznam rozdělen na sekce oddělené znakem |.
3. Z první sekce nás zajímá datum a ze čtvrté ID serveru.
4. Z páté sekce jsou načtena data obsahující samotný text záznamu. Následně je sekce rozdělena na dvě části - první slovo a zbytek věty.
5. Poté je pomocí větvení *SWITCH* na základě načteného prvního slova a následných vnořených větvení pomocí *SWITCH* zpracován zbytek věty a získány potřebné údaje.

3.3.3 RemoteAccess

Třída, která je spouštěná při startu a po spuštění očekává spojení od vzdáleného klienta BotAdmin. Třída pracuje s self-signed ssl certifikáty. Proto tento certifikát musí být načten v BotAdmin_remoteClient jako důvěryhodný. Vlastní self-signed certifikát si lze v JAVA vygenerovat pomocí příkazu `keytool -genkey -keyalg RSA -alias selfsigned -keystore <keystore> .jks -storepass <password> -validity 360 -keysize 2048` kde <keystore> je název souboru s certifikátem a <password> je heslo k certifikátu. Toto je třeba uložit do konfigurace BotAdminu.

Rozhodl jsem se načítat certifikát při startu programu z důvodu omezení úprav uložení certifikátů u uživatelů. Z tohoto důvodu musí být používán certifikát uložený na PC, kde běží BotAdmin_remoteClient. Informace a části kódu, které jsem použil při tvorbě této třídy jsem získal z [2].

3.3.4 prepare.sql

V tomto souboru jsou uloženy příkazy pro tvorbu tabulek v databázi programu BotAdmin. Jedná se o samostatné tabulky, které mezi sebou nejsou propojeny. Tabulky mají některá data společná, ale aby došlo k propojení primární/cizí klíč, bylo by třeba vytvořit ještě tabulku s uživateli a jejich ID. Jako nepraktické se mi jeví řešení vytvářet tabulku s daty, která jsou uložena již v DB TS3 Serveru. Zároveň by mohlo docházet k nekonzistenci dat mezi DB programu BotAdmin a TS3 Serveru. V následujících podsekcích popíšu vytvořené tabulky a jejich účel.

3.3.4.1 Actions V této tabulce jsou ukládány záznamy načtené z TS3 logu pomocí třídy LogCheck 3.3.2.

- time - čas záznamu
- targetType - typ cíle, něhož se záznam týká
- targetId - ID cíle
- targetAction - akce, jaká byla provedena nad cílem

- targetDbid - pokud je cílem zároveň uživatel, bude zde jeho ID
- invokerDbid - pokud akci vyvolal nějaký uživatel, tak zde bude jeho ID
- actionString - kompletní záznam z logu TS3 serveru zbavený speciálních znaků

3.3.4.2 channelConnected Zde jsou uloženy záznamy o přítomnosti uživatele v místnostech. Tyto záznamy jsou ukládány na základě notifikace ze strany TS3 Serveru.

- cid - ID kanálu, do kterého se uživatel připojil
- since - kdy se do kanálu připojil
- into - kdy kanál opustil
- cldbid - ID uživatele
- reasonId - důvod připojení do kanálu
- invokerCldbId - pokud se uživatel nepřipojil dobrovolně do kanálu, bude zde ID toho, kdo ho do kanálu dostal

3.3.4.3 IpAddress Zaznamenané IP adresy, ze kterých se uživatel připojil.

- ipAddress - IP adresa
- cldbid - ID uživatele
- count - počet, kolikrát se z dané IP adresy připojil

3.3.4.4 Nicks Zaznamenané přezdívky, které uživatel použil.

- nick - přezdívka uživatele
- cldbid - ID uživatele
- count - počet použití přezdívky. Nemusí souhlasit s počtem připojení. Přezdívka se ukládá i při editaci kanálu a dalších činnostech, protože přes TS3 ServerQuery nepřijde notifikace, při změně přezdívky uživatele.

3.3.4.5 serverInfo Zde se ukládají informace o stavu serveru.

- time - čas záznamu
- bandwidth_sent_last_minute - průměrný traffic za poslední minutu [B/s]
- bandwidth_received_last_minute - průměrný traffic za poslední minutu [B/s]
- packetlost - průměrná ztrátovost packetů [%]

- ping - průměrná odezva serveru [ms]
- client_count - počet připojených klientů celkem
- query_client_count - počet připojených query klientů

3.3.4.6 rules Zde jsou pravidla, které se mají provést na základě akcí uložených do Actions 3.3.4.1.

- ifTargetType - podmínka typu cíle
- ifTargetId - podmínka ID cíle
- ifTargetAction - podmínka akce vykonané na cíli
- RuleData - příkaz, který se má provést na základě splněných podmínek

3.3.4.7 timedRules Zde jsou pravidla, které se mají provést v určitých časových intervalech.

- firstStart - první provedení pravidla
- repeat - jak často se má pravidlo provádět (0=neopakovat)
- ruleData - příkaz, který se má provést na základě splněných podmínek

3.3.4.8 channelConnectedRule Zde jsou pravidla, které se mají provést po připojení uživatele v určité skupině do určitého kanálu.

- cid - ID kanálu
- sgid - ID serverové skupiny
- ruleData - příkaz, který se má provést na základě splněných podmínek

3.3.4.9 pmCanIssue Zde je seznam příkazů, který může provést určitá skupina pomocí chatu s query klientem přes TS3 klienta. Nekomoluje se výše pravomoci a zda má query klient práva na provedení tohoto příkazu.

- command - povolený příkaz
- sgid - ID serverové skupiny

3.3.5 SQLReader

Tato třída slouží k načtení SQL příkazů ze souboru. Používá se při inicializaci databáze a při požadavku provedení příkazů z SQL souboru při použití GUI. Kód jsem převzal z [3] a opravil chybu rozdělení triggerů do několika příkazů.

3.3.6 Utils

Třída poskytující podpůrné metody celému programu. Jsou zde často používané metody *escape* a *unescape* popsané v sekci 3.6.1.1.6.

Metoda *toMap*, používaná při práci s TS3 ServerQuery, kdy tato metoda uloží parametry odpovědi do HashMapy a tak se s těmito parametry poté pracuje lépe.

Dále je zde metoda *resultSetToTableModel*, která převede resultSet získaný z databáze do TableModelu vhodného pro zobrazení v GUI.

Další důležitá metoda je *levenshteinDistance*. Tuto metodu jsem převzal z [4]. Tato metoda vrátí seznam všech výskytů podřetězce v řetězci v zadané maximální vzdálenosti. Tato vzdálenost je definována jako maximální počet operací (vkládání, mazání a substituce) takových, aby byly řetězce totožné. Tato metoda se používá při hledání duplicitních registrací na serveru na základě podobnosti přezdívky uživatele.

3.3.7 TsQuery

Toto je jedna z nejdůležitějších tříd programu. Zajišťuje komunikaci programu s TS3 ServerQuery. Jsou zde metody pro navázání spojení (*prepare()*) a obnovu spojení při vzniklé chybě (*resetTsQuery()*). Dále jsou zde metody pro používané příkazy a startuje se zde vlákno *TsQueryReader* jehož funkce je popsána v sekci 3.3.8. Původně byla funkce *TsQueryReader* situována v tomto vláknu, ale protože dochází k předávání informací a třídění dat pomocí *PipedStream*, docházelo při zaplnění bufferu tohoto streamu k deadlocku.

3.3.8 TsQueryReader

Tato třída se stará o příjem dat od TS3 ServerQuery a jejich třídění do proudů *Notify* a *Answer*. Ze začátku jsem používal třídění podle dat, které posílá server. *Notify* zprávy začínají vždy *notify...*, ale během testování jsem zjistil, že občas i odpovědi na příkazy začínají *notify...*. Jediný rozdíl, jak tedy poznat jestli se jedná o *notify* je tedy, buď počkat na další řádek odpovědi, zda zde bude *error status* nebo si značit, že provádím příkaz a podle toho jednat. Pokud bych čekal na *error status*, tak by docházelo k zbytečnému zdržování reakce na *notify* a uváznutí programu do doby, než přijde další *notify* nebo provedu další příkaz.

Mnou zvolený přístup byl, že než začnu provádět příkaz nastavím status. Část kódu řešící tuto problematiku je ve výpisu programu 2. Nejdřív se zkontroluje, zda jsou načtena nějaká data. Pokud je řádek prázdný, tak se řádek dále nezpracovává nebo pokud to je podruhé za sebou, nejedná se o oddělovací řádek, ale o data odpovědi a uloží se. Pokud tedy *TsQueryReader* nějaká data načte, zkontroluje příznak. Pokud neprovádím příkaz, pošlou se data automaticky do *notify* proudu. Pokud provádím příkaz, podívám se jestli začíná na *error*. Pokud ano, vezmu předchozí řádek a zapíšu ho do *answer* proudu, zapíšu přijatý *error* řádek do *answer* proudu a vymažu příznak provádění příkazu. Pokud řádek nezačíná na *error*, uložím jej do pomocné proměnné a čtu další řádek. Pokud bude začínat tento řádek na *error* postupuji viz výše. Pokud nebude začínat na *error*, značí to situaci, že

ještě dřív než jsem dostal odpověď na zasláný příkaz, tak jsem přijal notifiy zprávu. Tedy řádek z pomocné proměnné zapíší do notifiy proudu a přijatý řádek uložím do pomocné proměnné a znovu čtu data od TS3 ServerQuery.

```

while (!stop && (line = query.getBr().readLine()) != null) {
    try {
        date = new Date();
        if (line.isEmpty()) {
            if (oldline.isEmpty()) {
                if (usableline != null) {
                    writeNotifyLine(date.getTime() + " " + usableline);
                    usableline = line;
                } else {
                    usableline = line;
                }
            } else {
                //zahodit prazdny radek
            }
        } else {
            if (query.isDoingCommand()) {
                if (line.startsWith("error")) {
                    if (usableline != null) {
                        answerOutputStream.write(usableline);
                        answerOutputStream.newLine();
                    }
                    answerOutputStream.write(line);
                    answerOutputStream.newLine();
                    answerOutputStream.flush();
                    usableline = null;
                    query.setDoingCommand(false);
                } else {
                    if (usableline == null) {
                        usableline = line;
                    } else {
                        writeNotifyLine(date.getTime() + " " + usableline);
                        usableline = line;
                    }
                }
            } else {
                writeNotifyLine(date.getTime() + " " + line);
            }
        }
        oldline = line;
    } catch (Exception ex) {
        if (!stop) {
            ex.printStackTrace();
        }
    }
}

```

Výpis 2: TsQueryReader - načítání dat

3.4 Podmínky běhu

1. Je potřeba přidat IP adresu PC, na kterém běží BotAdmin do souboru `query_ip_whitelist.txt`, který je ve složce, kde je uložen TS3 Server. BotAdmin posílá mnoho požadavků a tak pokud by v tomto listu nebyl, došlo by k zablokování přístupu.
2. Je potřeba spouštět server s parametrem `logappend=1`. Server tak bude pokračovat v logu do předcházejícího souboru. Pokud by nebyl tento parametr nastaven, došlo by při každém restartu serveru k vytvoření nového log souboru. Přes TS query lze přistupovat pouze k poslednímu vytvořenému souboru.
3. BotAdmin i BotAdmin.remote je potřeba spustit v JAVA VM s kódováním nastaveným na UTF8. Pokud tomu tak nebude, program bude sice fungovat, ale některé znaky se nezobrazí správně. Ve Windows se tato operace provede například nastavením globální proměnné `-Dfile.encoding=UTF8`.

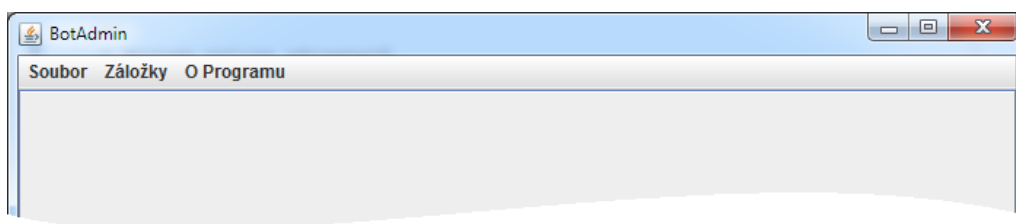
3.5 Instalace

Program nepotřebuje instalaci. Je dostačující nakopírovat potřebné soubory do libovolné složky a spustit v JAVA VM například příkazem `javaw.exe -jar BotAdmin.jar`.

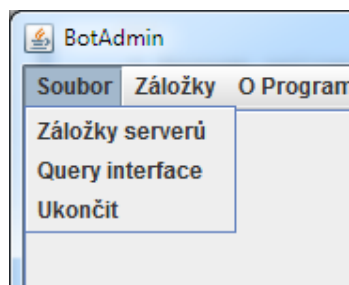
Soubory potřebné k běhu programu jsou uloženy v příloze na CD.

3.6 Ovládání

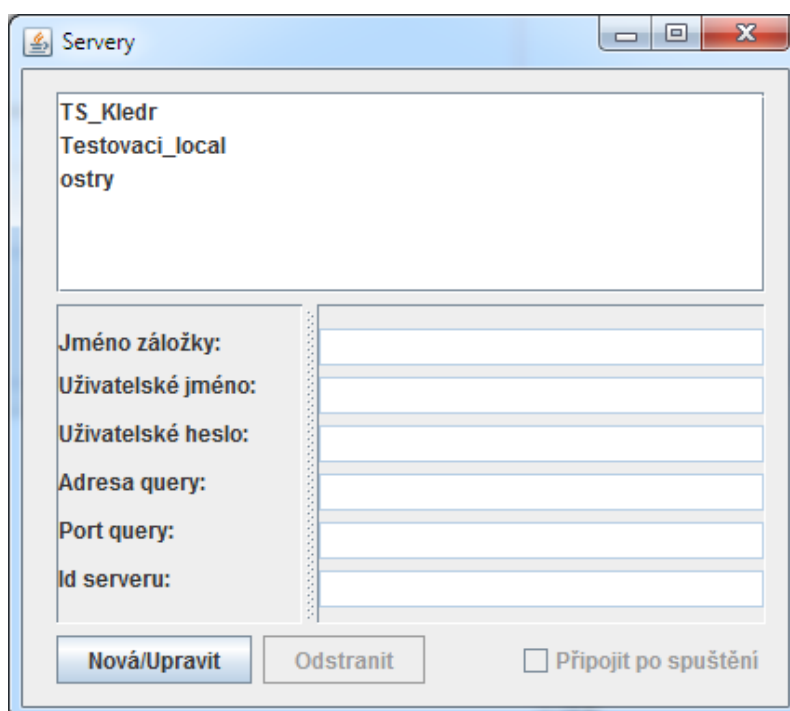
Okno spuštěného programu je zobrazeno na obrázku 7. Přes nabídku *Soubor/Záložky serverů* zobrazenou na obrázku 8 se dostaneme na okno (obrázek 9) s možností přidání a správy serverů. Jak si na tomto obrázku můžeme všimnout, záložka je identifikovaná svým jménem. Toto jméno je důležité také pro vzdálený přístup přes BotAdmin.remote. Přes něj se klienti připojují k BotAdmin a ten využívá data uložená v této záložce. Dále je třeba vyplnit uživatelské jméno a heslo pro přístup na TS3 ServerQuery, adresu a port, kde běží TS3 server a ID virtuálního serveru. Také je zde checkbox, který pokud je zatržen, tak se při spuštění programu automaticky spustí přihlášení podle záložky a program vykonává svou funkci.



Obrázek 7: BotAdmin - hlavní okno

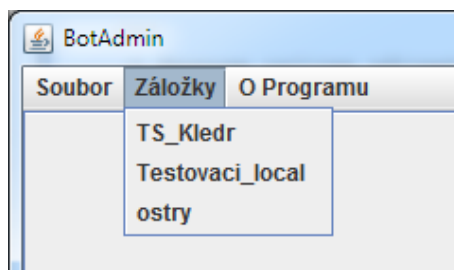


Obrázek 8: BotAdmin - nabídka Soubor

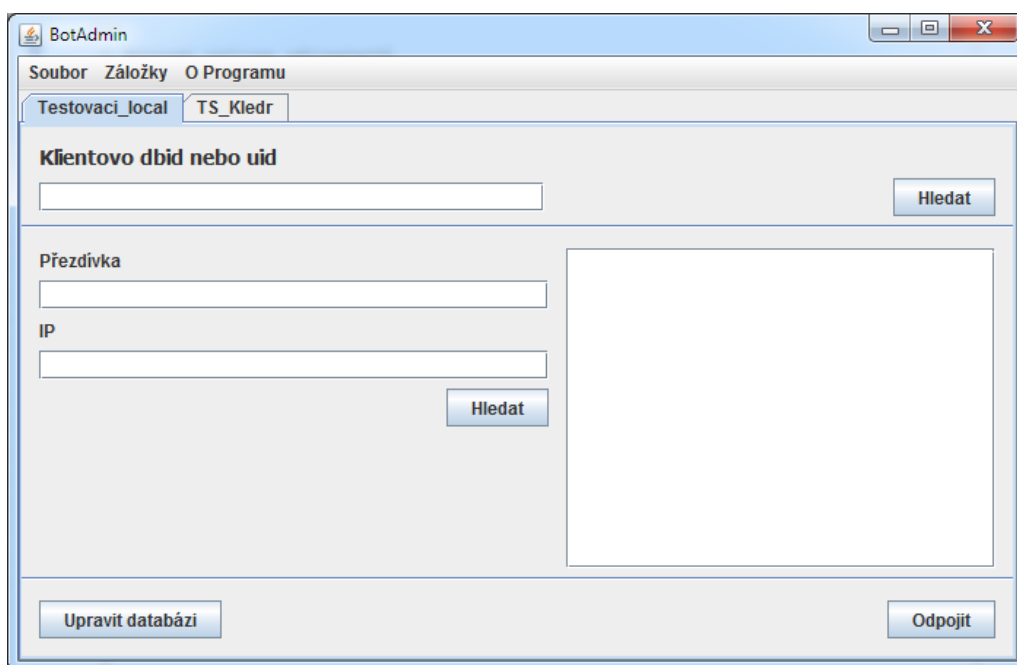


Obrázek 9: BotAdmin - úprava záložek

Pro spuštění programu do svého "akčního" běhu klikneme v hlavním menu na *Záložky* (obrázek 10) a pokud už máme záložky vytvořené, zobrazí se zde seznam uložených záložek. Pokud na některou klikneme, otevře se nová záložka v hlavním okně (obrázek 11).



Obrázek 10: BotAdmin - záložky

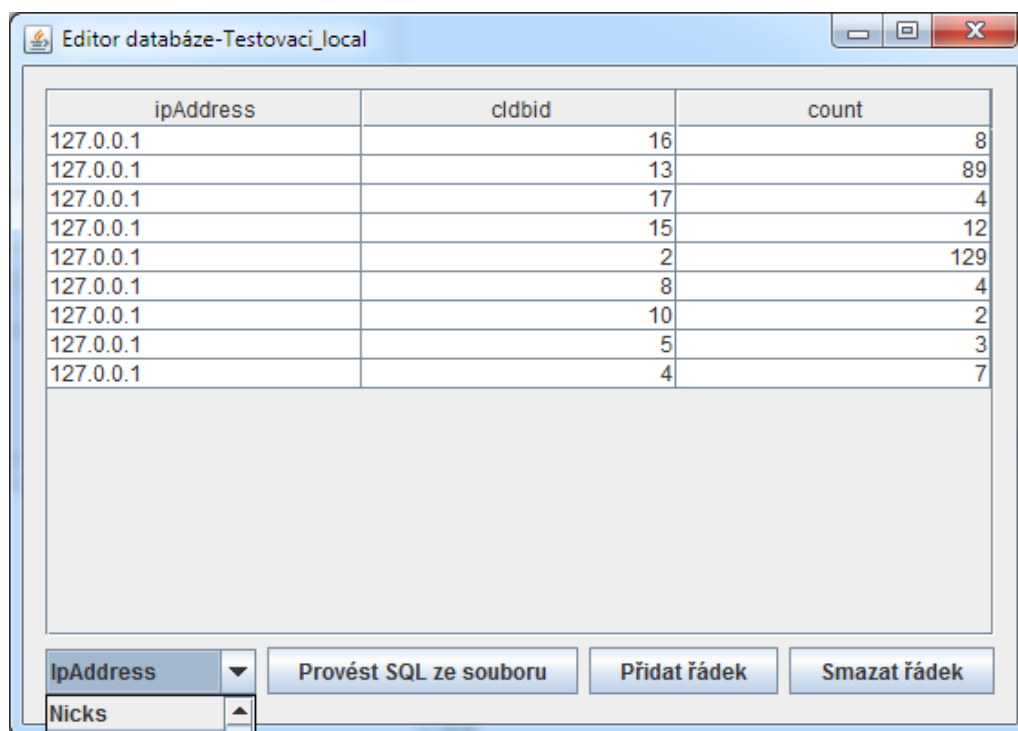


Obrázek 11: BotAdmin - spuštěné připojení na TS3 server

Poté už můžeme vyhledávat klienty podle jejich cldbídní nebo cluid. Taktéž můžeme vyhledávat klienty podle použitého nicku nebo IP adresy, ze které se klient někdy připojil. Ve vyhledávání lze použít zástupné znaky dle sql konvence. Tzn:

- % pro libovolný počet znaků
- _ pro jeden znak
- [a-z] pro znak z rozsahu a–z

- [abcd] pro znak *a* nebo *b* nebo *c* nebo *d*
- [^c] ne znak *c*



Obrázek 12: BotAdmin - úprava dat v databázi

Na obrázku 11 v pravém dolním rohu je pak tlačítko pro odpojení od TS3 Serveru. V levém dolním rohu je tlačítko pro úpravu lokální databáze, kterou si BotAdmin vytváří. Tato funkce není přístupná pro vzdálený přístup. Po kliknutí na tlačítko se otevře nové okno (obrázek 12) s možnostmi úprav. V tomto okně lze upravovat tabulky uložené v sqlite databázi otevřené záložky. Lze přidávat i mazat záznam, případně provést sql příkazy, které si připravíme v souboru.

Chceme-li vyhledat informace o nějakém konkrétním klientovi, tak na záložce serveru (obrázek 11) zadáme buď cldbid nebo cluid a stiskneme hledat. Otevře se nám okno s detailem o klientovi (obrázek 13). Pokud budeme chtít hledat uživatele podle přezdívky nebo IP adresy, vyplníme příslušné pole a stiskneme hledat. Pokud dojde ke shodě zobrazí se vpravo seznam možných cldbid uživatelů. Pokud dojde ke shodě jenom na základě IP adresy, bude vedle dbid napsáno IpAddress. Pokud dojde ke shodě i na základě přezdívky, bude tam tato přezdívka uvedena. Otevřít okno s detailem uživatele je poté možno buď dvojklikem na příslušný vyhledaný řádek nebo vyhledáním přes klientovo cldbid.

Detail uživatele

Unikátní identita: 9-VmptOYEGfMnPEsJY4PAoJX9c=

Databáze ID: 4377

Přezdívky

nick	Alternativní reg.	count
"CALISTA*BOJOWNICE"	51. [PIOV]CALISTA*BO	7
[PIOV]CALISTA*BOJOWNICE*	1640. IpAddress	33

IP Adresy

IpAddress	count
91.241.23.1	19

Akce

time	actionString
1363718371000/2013-03-19 18:39:31.312000/INFO	[VirtualServer] 1) client Br.potilasCZECH@id.4403) was added to channelgroup "Channel Admin"(id.5) by client [PIOV]CALISTA*BOJOWNICE"(id.4377) in channel "07. [PIOV]PIOVRA"(id.217)
1363803743000/2013-03-20 18:22:23.445000/INFO	[VirtualServer] 1) client TeamSpeakUser(id.4403) was added to channelgroup "Domácí"(id.14) by client [PIOV]CALISTA*BOJOWNICE"(id.4377) in channel "cspacat1">>> DarkOrbit MMO CZ1 <<<(id.2)
1363803799000/2013-03-20 18:23:19.213000/INFO	[VirtualServer] 1) client TeamSpeakUser(id.4403) was added to channelgroup "Channel Admin"(id.5) by client [PIOV]CALISTA*BOJOWNICE"(id.4377) in channel "07. [PIOV]PIOVRA"(id.217)
1363810195000/2013-03-20 20:09:55.695000/INFO	[VirtualServer] 1) client [MEC]e<OCEB>>>HEARTO>>>(id.614) was added to channelgroup "Channel Admin"(id.5) by client [PIOV]CALISTA*BOJOWNICE"(id.4377) in channel "07. [PIOV]PIOVRA"(id.217)
1363810825000/2013-03-20 20:25:59.2000/INFO	[VirtualServer] 1) channel "07. [PIOV]PIOVRA"(id.217), edited by [PIOV]CALISTA*BOJOWNICE"(id.4377)
1363892064000/2013-03-21 18:54:24.615000/INFO	[VirtualServer] 1) client [MEC]e<OCEB>>>HEARTO>>>(id.614) was added to channelgroup "Domácí"(id.14) by client [PIOV]CALISTA*BOJOWNICE"(id.4377) in channel "cspacat1">>> DarkOrbit MMO CZ1 <<<(id.2)

Jako cíl akce

time	actionString
1363450451000/2013-03-16 16:14:11.663000/INFO	[VirtualServer] 1) client [id.4377] was added to servergroup "Registrace"(id.7) by client [T/D]m-acspatpaprha20M-1(id.120)
1363450455000/2013-03-16 16:14:16.134000/INFO	[VirtualServer] 1) client [id.4377] was added to servergroup "MMO CZ1"(id.10) by client [T/D]m-acspatpaprha20M-1(id.120)
1363450460000/2013-03-16 16:14:20.518000/INFO	[VirtualServer] 1) client "CALISTA*BOJOWNICE"(id.4377) was added to channelgroup "Domácí"(id.14) by client [T/D]m-acspatpaprha20M-1(id.120)
1363450463000/2013-03-16 16:14:23.847000/INFO	[VirtualServer] 1) client "CALISTA*BOJOWNICE"(id.4377) was added to channelgroup "Domácí"(id.14) by client [T/D]m-acspatpaprha20M-1(id.120)
1363450466000/2013-03-16 16:14:26.115000/INFO	[VirtualServer] 1) client [id.4377] was added to servergroup "Astronauka"(id.53) by client [T/D]m-acspatpaprha20M-1(id.120)
1363450469000/2013-03-16 16:14:29.564000/INFO	[VirtualServer] 1) client [id.4377] was added to servergroup "Náčelník"(id.15) by client [T/D]m-acspatpaprha20M-1(id.120)
1363450480000/2013-03-16 16:14:40.409000/INFO	[VirtualServer] 1) client "CALISTA*BOJOWNICE"(id.4377) was added to channelgroup "NáčelníkZásupce"(id.11) by client [T/D]m-acspatpaprha20M-1(id.120) in channel "07. [PIOV]PIOVRA"(id.217)

Obrázek 13: BotAdmin - detail uživatele

Pokud se podíváme na detail klienta (obrázek 13), tak lze shora zleva doprava nalézt na prvním řádku cfluid a cldbid. Na druhém řádku je seznam s možnými alternativními registracemi klienta (stejně vyhledávání podle IP a následně porovnání na schodu nicku jako je popsáno výše) následované indikátorem průběhu načítání informací o klientovi. Dále zde najdeme bloky s informacemi o použitých přezdívkách a jejich počtu, o IP adresách ze kterých se uživatel připojil a kolikrát se z nich připojil. Dále zde můžeme vidět akce, které uživatel na serveru provedl a jako poslední zde nalezneme akce, kde byl uživatel příjemcem této akce.

3.6.1 Editace pravidel

Zde si popíšeme na jakém systému fungují pravidla, jak je přidávat a jaké parametry zde můžeme uvést.

3.6.1.1 Speciální příkazy V programu jsou vytvořeny příkazy, které kombinují několik příkazů z TS3 ServerQuery a doplňují tak funkce, které jednoduchým příkazem přes TS3 nejdou provést.

3.6.1.1.1 setclientchannelgroupincludsub Nastaví kanálovou skupinu v určeném kanálu a všech jeho podkanálech.

Parametry:

- cid=xx -> ID kanálu, ve kterém se má nastavit
- cgid=xx -> ID kanálové skupiny, kterou se má nastavit
- cldbid=xx -> ID uživatele, kterého se změna týká

3.6.1.1.2 sendmessagetoservergroup Odešle zprávu všem uživatelům, kteří jsou v určené skupině.

Parametry:

- msg=Ahoj\ssvětě!!! -> Zpráva, která má být poslána. Musí být zbavena speciálních znaků viz. 3.6.1.1.6
- sgid=xx -> ID serverové skupiny
- inclOffline=xx -> true pokud chceme poslat i offline zprávu uživatelům, kteří nejsou připojení.

3.6.1.1.3 serversnapshotcreatetofile Vytvoří zálohu serveru a tu uloží do textového souboru. Soubor bude označen časem pořízení a ID serveru. Soubor se uloží do složky, která je nastavena v konfiguračním souboru (defaultní /backups).

JMÉNO	ZNAK	ASCII	NAHRADIT ZNAKEM	NAHRADIT ASCII
Zpětné lomítko	\	92	\\	92 92
Lomítko	/	47	\ /	92 47
Mezera	" "	32	\ s	92 115
Roura		124	\ p	92 112
Zvonek	\ a	7	\ a	92 97
Návrat o znak	\ b	8	\ b	92 98
Další stránka	\ f	12	\ f	92 102
Nový řádek	\ n	10	\ n	92 110
Návrat vozíku	\ r	13	\ r	92 114
Tabulátor	\ t	9	\ t	92 116
Vertikální tabulátor	\ v	11	\ v	92 118

Tabulka 1: Tabulka escape znaků

3.6.1.1.4 servergrouppoke Odešle poke (popup zprávu) všem uživatelům, kteří jsou v určené skupině a jsou připojeni.

Parametry:

- msg=Ahoj\ssvětě!!! -> Zpráva, která má být poslaná. Musí být zbavena speciálních znaků viz.3.6.1.1.6
- sgid=xx -> ID serverové skupiny

3.6.1.1.5 addnote Přidá poznámku k uživateli. Uloží se do databáze (tabulky actions) jako každá jiná akce. Tato poznámka se vypisuje taky u příkazu 3.6.1.1.8 checkclient.

3.6.1.1.6 escape Převeďte text následující za příkazem do formátu vhodného k použití v parametrech příkazů. Např. text "Ahoj světe!!!" převede na "Ahoj\ssvětě!!!". Takto je třeba upravit text při používání vět s mezerami a jinými speciálními znaky. Kompletní seznam znaků je v tabulce 1. Tato úprava textu je důležitá, protože TS3 ServerQuery používá k oddělování parametrů příkazů tyto znaky. Pokud bychom úpravu neprovedli, může dojít k neprovedení příkazu nebo text příkazu nebude kompletní.

3.6.1.1.7 unescape Obrácená funkce k 3.6.1.1.6 escape.

3.6.1.1.8 checkclient Klient, který použije tento příkaz, dostane od BotAdmina textovou zprávu s rychlými informacemi o zjišťovaném klientovi. V této zprávě je obsažena jeho poslední použitá přezdívka, kolikrát byl vyhozen (jak z místnosti, tak ze serveru), kolik dostal banů, možné alternativní registrace a pokud byla uložena nějaká poznámka o uživateli pomocí příkazu addnote 3.6.1.1.5, tak i tato poznámka. Za příkazem následuje jeden z následujících parametrů.

Parametry:

- cldbid -> cldbid zjišťovaného klienta
- cluid -> cluid zjišťovaného klienta
- "" -> pokud za příkazem checklient neuvedeme nic, dojde ke zjištění informací o všech klientech, kteří jsou připojeni ve stejné místnosti jako uživatel, který tento příkaz zaslal.

3.6.1.2 Speciální parametry příkazů Zde je pár parametrů, které lze využít při tvoření pravidel pro BotAdmin. Příkaz, který se má provést je vždy prohledán na výskyt těchto řetězců. Pokud dojde k jeho výskytu je příslušně nahrazen.

3.6.1.2.1 <targetType> Tento řetězec je nahrazen číslem, které odpovídá typu zdroje, který vyvolal akci.

- klient=1
- kanál=2
- server=3
- kanálová skupina=5
- serverová skupina=6
- pravomoc=7

3.6.1.2.2 <targetId> Tento řetězec je nahrazen číslem, které odpovídá ID zdroje, který vyvolal akci.

- klient -> cldbid
- kanál -> cid
- server -> sid
- kanálová skupina -> cgid
- serverová skupina -> sgid
- pravomoc -> pid

3.6.1.2.3 <targetAction> Tento řetězec je nahrazen číslem, které odpovídá akci zdroje, který vyvolal akci.

- přidání=1
- odstranění=2
- editace=3

3.6.1.2.4 <targetCldbid> Tento řetězec je nahrazen číslem, které odpovídá cldbid uživatele, který byl příjemcem akce, která vyvolala námi tvořené pravidlo. Např. pokud je uživatel přidán do skupiny, bude tam cldbid uživatele, který byl přidán do této skupiny.

3.6.1.2.5 <invokerCldbid> Tento řetězec je nahrazen číslem, které odpovídá cldbid uživatele, který byl původcem akce, která vyvolala námi tvořené pravidlo. Např. pokud je uživatel přidán do skupiny, bude tam cldbid uživatele, který přidal tohoto uživatele do této skupiny.

3.6.1.2.6 <targetCluid> Stejně jako v 3.6.1.2.4 s tím rozdílem, že bude vloženo cluid místo cldbid.

3.6.1.2.7 <invokerCluid> Stejně jako v 3.6.1.2.5 s tím rozdílem, že bude vloženo cluid místo cldbid.

3.6.1.2.8 <targetClid> Stejně jako v 3.6.1.2.4 s tím rozdílem, že bude vloženo clid místo cldbid. Pokud uživatel nebude online, bude vloženo -1.

3.6.1.2.9 <invokerClid> Stejně jako v 3.6.1.2.5 s tím rozdílem, že bude vloženo clid místo cldbid. Pokud uživatel nebude online, bude vloženo -1.

3.6.1.3 Časovaná pravidla Otevřeme v editoru databáze (obrázek 12) a v levém dolním rohu si vybereme z rozevíracího seznamu tabulku *timedRules*. Máme zde tři sloupce. V prvním sloupci (firstStart) je čas prvního spuštění. V druhém sloupci (repeat) je, jak často se má akce opakovat. Obojí je čas v milisekundách. Čas prvního spuštění se počítá standardně od 1.1. 1970, 00:00:00 GMT. Také se zde uvádí GMT čas. V posledním sloupci (ruleData) je příkaz, který se má provést. Lze zde dát jakýkoli příkaz z TS3 ServerQuery plus příkazy zmíněné v 3.6.1.1. Taktéž zde můžeme využít parametrů, které jsou zmíněny v sekci 3.6.1.2. Pouze musíme počítat s tím, že všechny parametry jsou nastaveny na hodnotu -1. Jejich použití se tedy nedoporučuje. Pokud dojde k přidání časového pravidla, tak se pravidla nezačnou vykonávat ihned. Je potřeba ukončit spojení odpojením od TS3 Serveru a nové přihlášení na server.

3.6.1.4 Pravidla založená na akcích Otevřeme v editoru databáze (obrázek 12) a v levém dolním rohu si vybereme z rozevíracího seznamu tabulku *rules*. Tato pravidla se vykonávají na základě načítání informací z logu TS Serveru. Pokud tedy je nastaveno načítání dat z logu každých 30 sekund, tak se provedou pravidla, pokud vyhovují podmínky až za tuto dobu. Po přidání těchto pravidel není třeba restartovat program. Pokud bude uloženo do DB, provede se při následujícím čtení dat z logu. Lze zde provádět jak příkazy z TS3 ServerQuery, tak příkazy uvedené v 3.6.1.1. Plně lze využít speciálních parametrů pro lepší interakci uvedenou v 3.6.1.2.

Příkaz, který se má provést se uloží do sloupce RuleData. Podmínky pro provedení pravidla se specifikují pomocí parametrů *ifTargetType*, *ifTargetId* a *ifTargetAction*. Jedná se o číselné konstanty, jejichž seznam a význam je uveden v 3.6.1.2.1, 3.6.1.2.2, 3.6.1.2.3. Kompletní seznam lze také najít v souboru *constants.properties*.

3.6.1.5 Pravidla založená na připojení uživatele do kanálu. Otevřeme v editoru databáze (obrázek 12) a v levém dolním rohu si vybereme z rozevíracího seznamu tabulku *channelConnectedRules*. Tato pravidla se kontrolují pokaždé, dojde-li k připojení/vstupu uživatele do místnosti. Pokud odpovídá ID kanálu, do kterého se uživatel připojil a je ve skupině odpovídající nějakému pravidlu, provede se toto pravidlo. U pravidla je nastaveno *targetCldbid* uživatele a *invokerCldbid* uživatele, pokud připojení nebylo dobrovolné. Lze tak použít jakýkoli příkaz z TS3 ServerQuery plus příkazy zmíněné v 3.6.1.1 s parametry, které jsou zmíněny v sekci 3.6.1.2.

Do *cid* se vyplňuje ID kanálu, který odpovídá pravidlu a do *sgid* se vyplňuje serverová skupina, na kterou se má reagovat. Do položky *ruleData* se ukládá příkaz, který se má provést.

3.6.2 Příklady pravidel

Zde si uvedeme příklady pravidel, způsob jejich zápisu a vykonávanou funkci.

3.6.2.1 Časovaná pravidla V tabulce 2 jsou uvedeny příklady časovaných pravidel.

V prvním řádku tabulky je pravidlo, které od *Neděle, 3. březen 2013 18:56:00 GMT* posílá co hodinu popup správu s informací všem uživatelům, co jsou v serverové skupině s ID 61 a jsou online.

V druhém řádku tabulky je pravidlo, které od *Čtvrtek, 11. duben 2013 12:00:00 GMT* každý den vymaže heslo na místnosti s ID 275.

firstStart	repeat	ruleData
1362336960000	3600000	servergrouppoke msg=Blíží\sse\skonec\saukce. sgid=61
1365681600000	86400000	channeledit cid=275 channel_password

Tabulka 2: Příklady časovaných pravidel

3.6.2.2 Pravidla založena na akcích V tabulce 3 jsou pravidla, která reagují na akce jiných uživatelů.

V prvním řádku tabulky je pravidlo, které reaguje na přidání (ifTargetAction=1) nějakého uživatele do serverové skupiny (ifTargetType=6) s ID 26 (ifTargetId=26). Bude poslána uživateli zpráva, který dal tohoto klienta do této serverové skupiny, s informací, že je požadováno přidání poznámky o přidání klienta do skupiny. Je zde přímo příkaz, který má poslat, aby došlo k přidání poznámky.

V druhém řádku tabulky je pravidlo, které reaguje na přidání (ifTargetAction=1) uživatele do serverové skupiny (ifTargetType=6) s ID 10 (ifTargetId=10). Tento klient bude také automaticky přidán do kanálové skupiny 14 v kanálu 2.

Ve třetím řádku tabulky je pravidlo, které reaguje na odebrání (ifTargetAction=2) uživatele z serverové skupiny (ifTargetType=6) s ID 10 (ifTargetId=10). Tomuto klientovi bude nastavena kanálová skupina 8 v kanálu 2 a všech jeho podkanálech.

ifTargetType	ifTargetId	ifTargetAction	RuleData
6	26	1	sendtextmessage targetmode=1 target=<invokerClid> msg=Pošli\s BotAdminovi\sdůvod\sudělení\sžluté\s karty\sve\stvaru\s-> \saddnote\s targetType=1\s targetId=<targetId> \s targetCldbid=<targetCldbid> \smsg=SEMnapisDUVOD
6	10	1	setclientchannelgroup cgid=14 cid=2 cldbid=<targetCldbid>
6	10	2	setclientchannelgroupincludsub cgid=8 cid=2 cldbid=<targetCldbid>

Tabulka 3: Příklady pravidel založených na akcích

3.6.2.3 Pravidla založena na připojení uživatele do kanálu V tabulce 4 jsou pravidla, která se vykonají, pokud vstoupí uživatel do sledované místnosti a existují pro ni pravidla.

Na prvním řádku je pravidlo, které odešle online správu uživateli, který se připojí do kanálu s ID 1 a je v serverové skupině s ID 8.

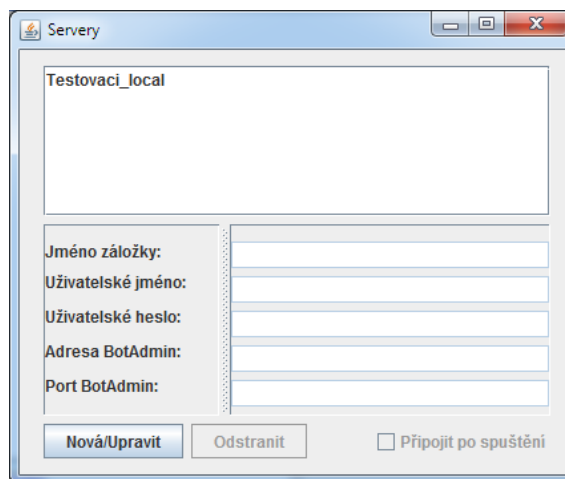
Na druhém řádku je pravidlo, které nastaví po připojení do místnosti (ID 285) uživateli (pokud je v serverové skupině 7) kanálovou skupinu s ID 14.

cid	sgid	ruleData
1	8	sendtextmessage targetmode=1 target=<targetClid> msg=Ahoj,\svítame\stě\szde\sna\snašem\sserveru. sgid=61
285	7	setclientchannelgroup cgid=14 cid=285 cldbid=<targetCldbid>

Tabulka 4: Příklady pravidel založených na připojení uživatele do kanálu

3.7 BotAdmin_remoteClient

Tento program slouží ke vzdálenému získávání dat o uživateli z programu BotAdmin. Program sám o sobě není schopen zapisovat žádná data do databáze ani TS3 Query. K jeho funkčnosti je potřeba, aby byl spuštěn program BotAdmin, na který se tento klient připojí. BotAdmin_remoteClient poté předá jen požadavek, co by potřeboval BotAdmin_remoteClient zjistit a o zbytek se postará BotAdmin. Funkcí tohoto programu není aktivní zjišťování informací, ale pouze pasivní zprostředkovávání dat o uživateli.



Obrázek 14: BotAdmin_remoteClient - editace záložek

Na obrázku 14 je zobrazeno menu pro uložení informací potřebných pro připojení na BotAdmin. Funkčnost se neliší od přidání serverů pro BotAdmin, pouze některé textové popisky byly upraveny.

K připojení je třeba znát IP adresu serveru, na kterém BotAdmin běží a port, na kterém přijímá požadavky od tohoto vzdáleného klienta. Dále je nutné pojmenovat záložku stejně, jako se jmenuje záložka spuštěná v programu BotAdmin, protože při načítání dat se z této záložky načítá adresa a port pro TS3 Query stejně jako ID virtuálního serveru, na který se program připojuje. Dále je třeba si vygenerovat přihlašovací údaje na TS3 Query a mít příslušné pravomoci pro provádění příkazů nad TS3 Query.

Tento podprogram využívá refraktorovanou kopii balíčku languages, support a UI hlavního programu BotAdmin. V balíčku *botadmin_remoteclient* jsou přítomny pouze lehce upravené třídy z balíčku *botadmin* hlavního programu potřebné k nastartování GUI a přesměrování požadavků na dotaz na DB a TS3 query do spojení na BotAdmin. Třídy jsou ochuzeny o vkládání informací do databáze a další vlákna, která nějakým způsobem aktivně zjišťují informace o klientech a pracují s těmito daty.

3.8 Testovací provoz programu

Program byl testován na domácím serveru provozovaném na Windows Vista 64bit (Atom D525@1.80Ghz, 4GB RAM). Server běží 24/7 v týdnu. Bylo tak možné testovat stabilitu programu a případné výpadky spojení a restarty serveru. Na tomto serveru zároveň běží také TS3 Server, na kterém je program testován. Během dne je na TS3 serveru připojeno průměrně 130 uživatelů. Prostor pro otestování stability a schopnosti reagovat na provoz dostatečně rychle byl zajištěn.

V počátečním stádiu vývoje programu byly problémy s obnovením činnosti po výpadku spojení s TS3 Serverem, což se stávalo zejména při pravidelné záloze TS3 Serveru a pravidelných plánovaných restartech serveru. Tato záloha byla nastavena stylem vypnutí serveru, záloha celé složky do *.rar archivu a jeho nové spuštění. Tyto zálohy byly z pohledu běžného uživatele nepříjemné, protože došlo k odpojení všech klientů a podle množství dat uložených na serveru docházelo i k 20 minutovému výpadku. Proto byly tyto zálohy nahrazeny zálohami pomocí programu BotAdmin. Tímto krokem nedojde k vypnutí serveru a zálohy proběhnou vpořádku. Nicméně bylo třeba v programu ošetřit i tyto nenadálé výpadky spojení a případy, kdy se BotAdmin spustí dříve než samotný TS3 Server. Toto je ošetřeno voláním metody reset...() příslušného vlákna, pokud dojde k vyvolání výjimky a není požadováno ukončení programu ze strany uživatele.

Dále bylo na serveru nastaveno cca 30 pravidel a v době psaní textu se pravidla rozrůstají. Během testování jsem nezaznamenal, že by pravidla způsobovala na serveru problémy či nestabilitu. Pouze u pravidel, která žádají interakci s více uživateli trvají delší dobu. Například zaslání online zprávy přihlášeným uživatelům a stejně tak offline zprávy ostatním nepřipojeným uživatelům, trvá odeslání zpráv cca 40s (při počtu 3200 uživatelů). Toto zpoždění je způsobeno nutností se dotázat TS3 Serveru na data každého uživatele zvlášť. Tento čas by se mohl zkrátit tvorbou vlastní DB uživatelů, nicméně by mohlo docházet k nekonzistenci dat uložených v DB BotAdmina a TS3 Serveru.

Při týdenním běhu aplikace alokuje cca 600MB v operační paměti. Dle mého názoru je toto číslo dost vysoké. Jako příčinu vidím únik paměti při běhu programu. Nicméně v době psaní práce jsem nebyl schopen zjistit, kde k tomuto úniku dochází. Při spuštění aplikace alokuje cca 40MB paměti.

4 Závěr

Program, který jsem napsal je funkční a provozu schopný. Pro svůj běh sice vyžaduje lehkou úpravu parametrů TS3 serveru, ale výhodné vlastnosti, které přinese tuto námahu vynahradí. Používáním programu na testovaném serveru klesly nároky na administrátory serveru a zlepšila se schopnost řešení problémů, které způsobují někteří problémoví uživatelé. Díky automatickým pravidlům se na server přidaly akce, které jiné doplňky neumožňují nebo vyžadují složitou konfiguraci na straně klienta. Používáním programu se také snížila čekací doba potřebná pro registraci na serveru, protože registraci pro příslušnou sekci dokončuje program BotAdmin. Z tohoto důvodu může provést registraci i administrátor cizí sekce, který nemá přístup do příslušné sekce. Během testování se snížila vytíženost administrátorů a také se snížil celkový počet administrátorů nutných pro běh serveru.

Program měl umožnit také detekci uživatelů se špatným nebo nestabilním spojením. Vzhledem k tomu, že TS3 Server poskytuje pouze informaci o tom, že uživatel ztratil spojení se tato funkce stala za daných okolností nevyužitelnou. Uživatelé, kteří mají tyto problémy a kontaktoval jsem je, většinou o problému vědí. Buď nechtěli pomoci s řešením nebo vyřešit problém nebylo v mých silách z důvodu problému na spojení mezi ISP. Z těchto důvodů nakonec tato funkce nebyla implementována.

Protože program dokáže poměrně rychle vyhledat informace o klientovi, byl splněn požadavek na umožnění zpětného dohledání informací o klientech a odhadnout tak jeho chování. Je umožněno sledovat akce klienta i ukládat poznámky o jeho chování. Tímto se zvýšila schopnost administrátorů řešit konflikty mezi klienty na základě předchozích poznámek, banů a stížností. Také bylo tímto programem umožněno sledování efektivity práce administrátorů a umožnit tak jejich případné odměňování či odebrání pravomocí pro neaktivitu.

Díky záznamu použitých přezdívek uživatelů a IP adres, ze kterých bylo přistupováno, byl také splněn požadavek na vyhledání duplicitních registrací. Také díky vytvoření univerzálního systému pravidel a přidání příkazů, které rozšiřují příkazy TS3 ServerQuery, bylo umožněno jednoduché zálohování serveru bez nutnosti odstávky serveru a případná jednoduchá obnova z této zálohy.

Také byla přidána jednoduchá konzola s nápovědou k TS3 příkazům s možností budoucího rozšíření o různé složené příkazy, které ulehčí práci s TS3 ServerQuery i nezkušeným uživatelům a administrátorům.

V budoucnu by bylo možné přepracovat BotAdmin_remote do podoby webového klienta, aby byl přístup komfortnější a nebylo nutné mít program uložený na přenosném médiu či v PC. Také je nutné najít a ošetřit úniky paměti, které se nepodařilo nalézt a opravit. Během provozu se také ukládají informace, které nelze získat přímo přes GUI programu ani jiným způsobem. V plánu na rozšíření programu je vytvoření webové služby, přes kterou by byla možnost získat stav serveru, jeho vytíženost a další informace o serveru. Rovněž se ukládá informace o navštívených místnostech, která ale není využívána. V budoucnu by se dala vytvořit funkce, která by hlídala vytíženost místností a v případě nízké vytíženosti informovala administrátory, nebo místnosti automaticky mazala. V neposlední řadě by se program mohl přepracovat na použití s plnohodnotnou

databází, se kterou je schopen pracovat také TS3 Server. Tímto lze získat přístup přímo do DB TS3 Serveru a mnoho informací by nebylo třeba získávat přes TS3 ServerQuery. Výsledkem by bylo zrychlení práce programu. Zároveň by se ale ztížila instalace programu a nároky na schopnost nastavení TS3 Serveru, což by mohlo odradit administrátory při použití programu na serveru.

Tomáš Juříčka

5 Reference

- [1] TEAMSPEAK SYSTEMS GMBH. *TEAMSPEAK 3 SERVERQUERY MANUAL* [online]. 2012-02-29 [cit. 2013-03-16]. Dostupné z: http://media.teamspeak.com/ts3_literature/TeamSpeak3ServerQueryManual.pdf
- [2] OWASP Foundation. *OWASP Java Project: Using the Java Secure Socket Extensions* [online]. 2010-02-21 [cit. 2013-03-16]. Dostupné z: https://www.owasp.org/index.php/Using_the_Java_Secure_Socket_Extensions
- [3] wer, Flo. *Coderanch: Execute .sql file using java* [online]. 2012-05-17 [cit. 2013-03-16]. Dostupné z: <http://www.coderanch.com/t/306966/JDBC/databases/Execute-sql-file-java>
- [4] Mička, Pavel. *Algoritmy.net: Levenshteinova vzdálenost* [online]. [cit. 2013-03-16]. Dostupné z: <http://www.algoritmy.net/article/1699/Levenshteinova-vzdalenost>